

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра автоматики та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Програмне забезпечення  
інформаційно-комунікаційних систем»  
спеціальності 121 «Інженерія програмного забезпечення»  
на тему: «Автоматизована система пошуку роботи»**

Виконав:

студент IV курсу, групи ІТ-61

Савінов Ілля Андрійович \_\_\_\_\_

Керівник:

Асистент кафедри АУТС

Шинкевич Микола Костянтинович \_\_\_\_\_

Рецензент:

\_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматики та управління в технічних системах**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Савінов Ілля Андрійович**

1. Тема проєкту «Автоматизована платформа пошуку роботи», керівник проєкту асистент Шинкевич Микола Костянтинович, затверджені наказом по університету від «\_07\_» \_травня\_ 2020 р. № 1081-с

2. Термін подання студентом проєкту 09.06.2020

3. Вихідні дані до проєкту

Автоматизована платформа пошуку роботи написана на мові JavaScript з використанням Node.js, React Native, середовища Visual Studio Code та базою даних MongoDB.

4. Зміст пояснювальної записки

1. Вступ 2. Опис предметної області 3. Аналіз існуючих рішень 4. Аналіз вимог до програмного забезпечення 5. Вибір технологій розробки 6. Розробка системи 7. Тестування програмного забезпечення 8. Впровадження та використання розробки системи.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

Діаграма варіантів використання, діаграма компонентів, діаграма послідовностей, діаграма структури бази даних.

6. Дата видачі завдання 22.02.2020

Календарний план

№ з / п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Приміт ка
1	Вибір тематичного напрямку та узгодження теми дипломного проекту	22.02.2020	
2	Аналіз теоретичних матеріалів та вивчення предметної області	15.04.2020	
3	Розробка технічного завдання, вибір методів та засобів реалізації задачі	24.04.2020	
4	Огляд існуючих рішень з тематики роботи	27.04.2020	
5	Розробка структури прототипу та проектування системи	06.05.2020	
6	Реалізація проекту	14.05.2020	
7	Налагодження та перевірка програми	23.05.2020	
8	Оформлення пояснювальної записки	25.05.2020	
9	Передзахист дипломного проекту	26.05.2020	
10	Доопрацювання пояснювальної записки та підготовка презентації	08.06.2020	
11	Захист дипломного проекту	17.06.2020	

Студент

Ілля САВІНОВ

Керівник

Микола ШИНКЕВИЧ

## **АНОТАЦІЯ**

Савінов І. А. Автоматизована платформа пошуку роботи. КПІ ім. Ігоря Сікорського, Київ, 2020. Ключові слова: JavaScript, ReactNative, платформа для пошуку роботи.

Основна частина документу викладена у пояснювальній записці, виконаній на 75 сторінках, та містить 3 рисунки та 54 таблиці.

Об'єктом розробки є платформа для пошуку роботи.

Мета розробки – розширення існуючих аналогів серед проектів для пошуку роботи.

У дипломному проекті проведено аналіз існуючих рішень, поставлені функціональні та нефункціональні вимоги, ретельно підібрані технології для розробки та спроектовано базу даних. Також розроблено програмний продукт та проведено уважне тестування системи.

Отримані результати можуть бути корисні при створенні аналогічних систем.

## **SUMMARY**

Savinov I. A. Automated job search platform. Igor Sikorsky KPI, Kyiv, 2020. Keywords: JavaScript, ReactNative, job search platform.

The main part of the document is set out in an explanatory note, made on 75 pages, and contains 3 figures and 54 tables.

The object of development is a job search platform.

The purpose of the development is to expand the existing analogues among job search projects.

The diploma project analyzes the existing solutions, sets functional and non-functional requirements, carefully selected technologies for development and designs a database. A software product was also developed and the system was carefully tested.

The results obtained can be useful in creating similar systems.

Номер рядка	Формат	Позначення	Найменування	Кіл. листів	№ екз.	Примітка		
1			Документація загальна					
2								
3			Знову зроблена					
4								
5	A4	IT61.170БАК.004 ПЗ	Автоматизована платформа	75				
6			пошуку роботи.					
7	A3		Пояснювальна записка.					
8		IT61.160БАК.004 Д1	Автоматизована платформа	1				
9			пошуку роботи.					
10			Діаграма варіантів					
11			використання.					
12	A3	IT61.160БАК.004 Д2	Автоматизована платформа	1				
13			пошуку роботи.					
14			Діаграма компонентів.					
15	A3	IT61.160БАК.004 Д3	Автоматизована платформа	1				
16			пошуку роботи.					
17			Діаграма структури					
18			бази даних.					
19	A3	IT61.170БАК.004 Д4	Автоматизована платформа	1				
20			пошуку роботи.					
21			Діаграма послідовностей.					
22								
23								
24								
25								
Змн.	Арк.	№ докум.	Підпис	Дата	IT61.170БАК.004 ТП			
Розроб.		Савінов І. А.			Автоматизована платформа пошуку роботи. Відомість технічного проєкту.	Літ.	Арк.	Акрушіє
Перевір.		Шинкевич М.К.					1	1
Реценз.						КПІ ім. Ігоря Сікорського ФІОТ. зр. IT-61		
Н. Контр.								
Затверд.								

**Пояснювальна записка  
до дипломного проєкту  
на тему: «Автоматизована платформа пошуку  
роботи»**

Київ – 2020

## ЗМІСТ

ВСТУП.....	5
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Обґрунтування доцільності розробки .....	7
1.1.1 Опис та аналіз предметного середовища .....	7
1.1.2 Аналіз функціональних особливостей системи .....	9
1.2 Призначення розробки .....	10
1.3 Цілі та задачі розробки.....	10
1.4 Висновок до розділу .....	10
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ .....	12
2.1 Існуючі платформи для пошуку роботи.....	12
2.1.1 Work.ua .....	13
2.1.2 Rabota.ua.....	14
2.1.3 Jooble .....	14
2.2 Аналіз і порівняння .....	15
2.3 Висновок до розділу .....	16
3 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	17
3.1 Сценарії використання системи.....	17
3.2 Розробка функціональних вимог до системи.....	36
3.3 Розробка нефункціональних вимог до системи .....	37
3.4 Висновок до розділу .....	38
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ.....	39

					IT51.230БАК.002 ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Савінов І. А.			Автоматизована платформа пошуку роботи. Пояснювальна записка			Лім.	Арк.	Акрушів	
Перевір.		Шинкевич М. К.								2	75
Реценз.								КПІ ім. Ігоря Сікорського ФІОТ, гр. ІТ-61			
Н. Контр.											
Затверд.											

4.1 Переваги вибраних технологій розробки .....	39
4.1.1 JavaScript .....	40
4.1.2 Node.js .....	41
4.1.3 MongoDB.....	42
4.1.4 React Native .....	43
4.1.5 Socket.io .....	44
4.1.6 Material UI .....	44
4.1.7 Ramda .....	45
4.2 Висновок до розділу .....	45
5 РОЗРОБКА СИСТЕМИ.....	46
5.1 Структура проекту.....	46
5.1.1 Структура серверної частини проекту.....	46
5.1.2 Структура клієнтської частини проекту .....	48
5.2 Структура обробки запитів .....	50
5.3 Розробка бази даних .....	51
5.4 Висновок до розділу .....	57
6 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	58
6.1 Фази життєвого циклу програмного забезпечення.....	58
6.2 Тестування програмного забезпечення .....	60
6.2.1 Ручне тестування .....	61
6.2.2 Автоматизоване тестування .....	61
6.4 Неперервна інтеграція .....	69
6.5 Висновок до розділу .....	70
7 ВПРОВАДЖЕННЯ ТА ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ..	71

					ІТ61.170БАК.004 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		



7.1 Апаратні та програмні вимоги для експлуатації програми .....	71
7.2 Інструкція з встановлення .....	72
7.3 Висновки до розділу .....	73
ВИСНОВКИ .....	74
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	75

					ІТ61.170БАК.004 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Майже у кожної людини настає час коли потрібно шукати заробіток або нову роботу. Тепер, коли у половини людства є доступ до Інтернету, все більше сфер людської діяльності переходить в режим онлайн. Майже кожна людина, організація чи навіть країна мають сторінку у Всесвітній мережі. Телефони та комп'ютери, які підключені до Інтернету вже не являються новинкою. Тепер ми підключаємо пристрої «розумного будинку», щоб з будь-якої точки планети керувати домівкою. Діджиталізація прагне щоб кожна людина могла вирішити всі свої проблеми діставши телефон з кишені.

Раніше, щоб знайти роботу, треба було купити новий випуск газети або уважно дивитися на дошки оголошень. Тепер дошки оголошень замінили на сайти для пошуку роботи, де кожна людина може знайти роботу не виходячи з дому. Ці сайти дають змогу компаніям чи організаціям публікувати вакансії, а потенціальним робітникам подавати заявки. Такі системи добре себе зарекомендували та набули популярності.

Оскільки є можливість більше автоматизувати процес пошуку, головною метою даного проекту є створення автоматизованої платформи пошуку роботи, для того, щоб надати користувачам альтернативне до існуючих рішень джерело вакансій.

Особистою метою розробки цього проекту являється поглиблення знань у сфері пошуку роботи та набути навички розробки програмного забезпечення.

Завданням цієї роботи є створення автоматизованої платформи пошуку роботи в сфері будівництва, де робітнику та компанії не потрібно витрачати час на пошук один одного, та надати це платформі. У ній буде присутня реєстрація та автентифікація компаній та робітників, публікація робіт, можливість для компаній шукати робітників, чати між компанією та робітником, аудіо-дзвінки.

Ще одним завданням проекту є дотримання життєвого циклу розробки програмного забезпечення: аналіз існуючих рішень, постановка вимог,

					IT61.170БАК.004 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

проектування, розробка, тестування та експлуатація. Це має мінімізувати проблеми під час розробки та зекономити час.

Для досягнення мети було використано найпоширеніші методи дослідження: спостереження, порівняння, узагальнення, аналіз та синтез.

Практичне завдання даного проекту полягає в створенні автоматизованої платформи пошуку роботи, покращеної порівняно з існуючими рішеннями.

Бакалаврський проект включає в себе наступні розділи: вступ, основні розділи, висновки та список використаних джерел із 9 найменувань. Графічна частина складається із 4 креслеників формату А3. Загальний обсяг - 75 сторінок.

					ІТ61.170БАК.004 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Обґрунтування доцільності розробки

Більшість тих, хто шукає роботу, використовують Інтернет, а не витрачають час на кругообіг реклами в газеті. Онлайн-полювання на роботу може заощадити витрати, пов'язані з рекламою та за короткий час підключити вас до більшості роботодавців. Вам не потрібно їздити до офісів роботодавців, щоб запитати, чи є у них місця роботи. Ви можете шукати роботу у своєму географічному районі або вивчити можливості в наступному місті, по всій країні або в іншому куточку світу. Крім того, якщо ви розпочали пошук роботи на відстані, деякі роботодавці проводять відео-інтерв'ю, що полегшує процес.

Автоматизована система пошуку роботи полегшує цей процес. Можна зекономити час робітників та компаній на пошук один одного віддавши це завдання системі. Тобто автоматизувати процес підбору робітників до опублікованих вакансій на сайті.

### 1.1.1 Опис та аналіз предметного середовища

Система пошуку роботи – це програмне забезпечення, головною функцією якого є надання можливості компаніям та потенціальним робітникам знайти один одного.

Зазвичай такі системи будуються на основі клієнт-серверної архітектури, де клієнтська сторона виступає в ролі сайту або мобільного додатку. Клієнт-серверна архітектура складається з розподілених застосунків та передбачає обмін даними між ними.

Дана архітектура передбачає такі компоненти:

- сервери, які взаємодіють з базою даних та сторонніми сервісами, обробляють дані та надають їх клієнтській стороні;

					IT61.170БАК.004 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

- клієнти, які зазвичай містять графічний інтерфейс для взаємодії користувача із системою, надсилають запити до серверної сторони;
- мережа, яка зв'язує серверну та клієнтську частини.

Оскільки в цій системі можна виділити два типи користувачів, то її використання варто розглядати з двох сторін: зі сторони компанії та зі сторони робітника.

Оскільки така система передбачає автоматизований підбір, то деякі параметри для роботи та профілі робітників мають бути заповнені заздалегідь визначеними значеннями. Таким чином охопити всі сфери бізнесу дуже складно, тому було вирішено зосередитися на платформі пошуку роботи в сфері будівництва.

Головною перевагою для компанії являється економія часі на пошуку потенціальних робітників, цим займається платформа. Публікація вакансії теж не займе багато часу завдяки наявності заздалегідь визначеним значенням для її параметрів. Якщо компанія, перед тим як публікувати вакансію, хоче продивитися робітників (які будуть підібрані до даної вакансії), вона може скористатися пошуком та оглянути профілі робітників. Форма пошуку має аналогічні параметри до параметрів вакансії, тому якщо компанію задовільняє такий підбір, вона може конвертувати параметри пошуку у вакансію.

Перевагами для робітника є можливість детально охарактеризувати свої навички, вибрати існуючі у нього сертифікати зі списку. Також є можливість публікувати у профілі фотографії виконаних робіт, що підвищує шанс отримати бажану роботу. Є можливість вказати своє місце проживання та дистанцію, яку робітник готовий подолати до зазначеного у вакансії місця.

Також платформа передбачає можливість спілкуватися у чаті з потенціальним робітником або використовувати аудіо-дзвінок, що не потребує використання інших джерел зв'язку.

					IT61.170БАК.004 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.1.2 Аналіз функціональних особливостей системи

Автоматизована система пошуку роботи буде мати два типи користувачів: компанія та робітник.

У розпорядженні компанії такі розділи:

- початкова сторінка (дошка), яка містить інформацію про нові повідомлення та кнопки швидкого доступу;
- пошук робітників;
- розділ з публікованими роботами;
- розділ з чатами.

На початковій сторінці розміщені кнопки для швидкого доступу на сторінки створення вакансії та пошуку, лічильники нових повідомлень та оновлень стосовно виставлених робіт.

Робітник має доступ до таких розділів:

- початкова сторінка, яка буде містити нові повідомлення та кнопки швидкого доступу;
- розділ зі своїм профілем, який можна переглядати та редагувати;
- розділ з інформацією стосовно вакансій;
- розділ з чатами;

До функціональних особливостей проекту відноситься функціонал, який треба реалізувати:

- система реєстрації та автентифікації;
- пошук робітників;
- створення вакансії;
- спілкування за допомогою чату та аудіо-дзвінків;
- автоматизований підбір робітників до опублікованих вакансій.

					IT61.170БАК.004 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1.2 Призначення розробки

Розроблений продукт призначений для налагодження зв'язків між роботодавцями та робітниками. Система орієнтована на англomовні країни та може охопити територію великого міста.

## 1.3 Цілі та задачі розробки

Цілі розробки проекту можна поділити на дві категорії: для компаній і для робітників.

Головною ціллю створення проекту для компаній – прискорення процедури пошуку кадрів на відкриту вакансію та надання можливості шукати робітників серед зареєстрованих у системі.

Головною ціллю створення проекту для робітників – звільнити робітника від довгих та монотонних пошуків та підібрати роботу, яка підходить саме йому.

Варто зазначити, ще однією метою є надання можливості спілкування між потенціальним робітником та компанією відразу на платформі.

## 1.4 Висновок до розділу

Проаналізувавши використання сучасних технологій та систем та тенденцій розвитку можна зробити декілька висновків.

По-перше, Інтернет все більше впливає на повсякденне життя кожної людини та надалі цей вплив буде тільки зростати. Всесвітня мережа надає можливість спілкуватися, навчатися, працювати та розважатись незалежно від місця знаходження.

По-друге, збільшення користувачів Інтернету призводить до збільшення попиту робітників на відстані, тому все більше робітників та компаній обирають

					ІТ61.170БАК.004 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

головним джерелом пошуку Інтернет. Для цього і були створені сайти з пошуку роботи.

По-третє, існуючі рішення не є бездоганними та можуть бути покращенні та реалізовані по іншому. Можна автоматизувати ще більше процесів та полегшити пошук роботи.

Також варто зазначити, що все більше персональних даних обробляється нескінченною кількістю додатків та систем, тому треба звертати особливу увагу на захист персональних даних. Використовувати найсучасніші рішення та слідкувати за їх оновленням.

					ІТ61.170БАК.004 ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		



## 2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

### 2.1 Існуючі платформи для пошуку роботи

В Інтернеті доступний широкий спектр сайтів та додатків для пошуку роботи, тому знайти той, який відповідає потребам, не надто складно. Більшість пропонують різноманітні функції для пошуку вакансій, а також статті та інформацію про те, як найкраще провести пошук роботи. Ці послуги, як правило, безкоштовні, коли шукач роботи реєструється на веб-сайті.

Веб-сайти, що працюють, зазвичай поділяються на дві групи. Деякі з них – це сайти набору персоналу, які перераховують вакансії, і робочий подає заявку безпосередньо на компанію. Такі сайти збирають різні посилання на веб-сайти та агенції з підбору персоналу, що дозволяють:

- переглядати загальну картину ринку;
- знайти роботу або підробіток;
- переглянути підбір персоналу для сектору.

Другим типом сайту керують кадрові агенції. Вони орієнтовані на продаж і вимагають від шукачів роботи заповнити вакансії. Робітник реєструє свою заявку та / або резюме з ними і, можливо, поспілкуєтесь про тип роботи, яку шукаєте, щоб вони, в свою чергу, могли допомогти вам знайти відповідну роботу.

Переваги використання онлайн-сайтів набору персоналу полягають у тому, що консультанти з підбору персоналу - це фахівці, і можна безкоштовно реєструватись у них та користуватися їх контактами. Ви зможете дізнатися про відкриття вакансій і подати заявку на посади в будь-який час та з будь-якого комп'ютера, що має доступ до Інтернету.

Далі будуть розглянуті популярні сайти для пошуку роботи на території України.

					ІТ61.170БАК.004 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

### 2.1.1 Work.ua

Work.ua — сайт пошуку роботи в Україні, його щомісяця відвідують 3,8 млн людей, більше 155 тис. — розміщують вакансії. За даними на серпень 2017 року на сайті зібрано близько 2 млн резюме та 90 тисяч вакансій. [1]

Сайт було створено дніпровською компанією «Реактор» у 2006 році та на теперішній час нараховує 5 млн чоловік активних користувачів щомісяця. Спочатку прибуток розробникам приносили такі функції, як надання вакансії особливого статусу в результатах пошуку та продаж привілейованих аккаунтів.

Основними функціями для шукача являються:

- створення профілю та резюме;
- пошук вакансій;
- подання резюме на вибрану вакансію;
- сповіщення про нові вакансії електронною поштою.

Основні функції для роботодавців:

- публікація вакансії;
- доступ до бази даних резюме.

Також роботодавець за гроші має можливість виділити вакансію у результатах пошуку, розмістити логотип компанії на головній сторінці та придбати привілейований аккаунт з додатковим функціоналом у вигляді оновлення вакансій щодня та іншими.

Сайт має розділ «Ресурси» зі статтями, які можуть допомогти як роботодавцям так і шукачам. Платформа має телеграм бот та мобільний додаток. Сайт також публікує державні вакансії, що свідчить про довіру до ресурсу.

Основні переваги Work.ua:

- зрозумілий інтерфейс;
- швидкий доступ до пошуку;
- можливість розширеного пошуку;
- відображення середньої зарплати по знайденому результату.

					ІТ61.170БАК.004 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Основні недоліки Work.ua:

- обмежена безкоштовна кількість подачі резюме за добу;
- обмежена кількість одночасно публікованих вакансій.

### 2.1.2 Rabota.ua

Rabota.ua — єдиний job-сайт, який представляє Україну в мережі провідних світових сайтів The Network за такими критеріями, як кількість і якість вмісту. Щомісячна відвідуваність перевищує чотири млн. унікальних відвідувачів, а загальна складає понад 10 млн. [2]

Ресурс був створений у 2000 році та став першим онлайн-ресурсом для пошуку роботи на території України. З часом провідна у Центральній Європі польська компанія з інтернет-рекрутингу стає головним партнером та спонсором цього проекту. В цей же час стає партнером міжнародної рекрутингової мережі The Network.

Основні переваги Rabota.ua:

- розділ з рекомендованими роботами на основі унікального алгоритму;
- кількість рубрик вакансій;
- гнучкий пошук;
- можливість додавати до роботи теги, що значно покращую пошук;
- розділ з популярними професіями.

Основні недоліки Rabota.ua:

- велика конкуренція завдяки популярності ресурсу;
- обмежена кількість розміщених вакансій одночасно.

### 2.1.3 Jooble

Щоб детальніше проаналізувати сферу існуючих рішень розглянемо сайт-агрегатор для пошуку роботи. Агрегатори – системи, які збирають та подають в

					ІТ61.170БАК.004 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

одному місці інформацію з різних сайтів та ресурсів. На сьогодні проекти такого типу є дуже популярними, тому що кількість ресурсів з однією тематикою тільки зростає, а користувач хоче мати все в одному місці.

Jooble — це вертикальний пошуковик, який агрегує оголошення про вакансії з тисяч джоб-бордів, газет, кар'єрних сторінок підприємств і рекрутингових агентств. Jooble працює в 71 країнах і підтримує більш ніж 24 мови та знаходиться у ТОП-1000 найвідвідуваніших сайтів світу, а з грудня 2019 є другим найвідвідуванішим сайтом з пошуку роботи з 90 млн відвідувачів на місяць в середньому та більш ніж 950 млн відвідувачів протягом всього 2019 року. [3]

Система збирає інформацію з інших сайтів з пошуку роботи, газет, сторінок компаній, агенств та дозволяє. Користувач починає пошук з заповнення міста на назви вакансії, потім можна відфільтрувати вакансії за віддаленістю, датою, зарплатою. Сотні параметрів враховуються алгоритмом, щоб підібрати потрібні вакансії.

Основні переваги Jooble:

- велика кількість вакансій;
- унікальний алгоритм пошуку;
- постійна підтримка та оновлення;
- уніфікація даних.

Основні недоліки Jooble:

- скудний інтерфейс;
- тільки стандартний функціонал.

## 2.2 Аналіз і порівняння

Розглянуті проекти мають схожу схему автоматизації процесу. Зареєстровані компанії подають вакансії, робітники за допомогою пошуку та різноманітної фільтрації знаходять потрібні роботу і подають резюме. Загалом

					IT61.170БАК.004 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

вони мають однаковий інтерфейс користувача та додають накладають функціонал.

Проаналізувавши ці застосунки було визначено основні моменти, які треба врахувати при розробці, а саме:

- простий у використанні інтерфейс;
- гнучкий пошук робітників;
- можливість спілкування компанії та робітника відразу на платформі;
- захищеність даних.

### 2.3 Висновок до розділу

Для створення програмного продукту обов'язковим етапом являється аналіз аналогів. Треба детально оглянути, проаналізувати та виділити переваги та недоліки існуючих рішень. На основі цих даних можна створити конкурентноспроможний продукт, який оминє недоліки та буде мати унікальний функціонал, чим привабить клієнтів.

Були детально оглянуті існуючі рішення, виділені основні переваги на недоліки кожного. На основі цих даних було виведено аспекти, які повинні бути реалізовані в проекті.

					ІТ61.170БАК.004 ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Сценарії використання системи

У розробленому кресленику ІТ61.170БАК.004 Д1 наведена діаграма варіантів використання (діаграма прецедентів). В таблиці 3.1 наведено варіанти використання майбутньої системи.

Таблиця 3.1 – Варіанти використання системи

Шифр варіанту використання	Назва варіанту використання
UC-1	Перегляд меню авторизації
UC-2	Введення номеру телефону користувача
UC-3	Введення смс коду та авторизація
UC-3.1	Авторизація робітника та вхід до системи
UC-3.1.1	Перегляд доступних вакансій
UC-3.1.2	Прийняти вакансію
UC-3.1.3	Відхилити вакансію
UC-3.1.4	Написати роботодавцю.
UC-3.1.5	Перегляд чатів відповідних до прийнятих вакансій
UC-3.1.6	Вихід із системи
UC-3.1.7	Перегляд профілю
UC-3.1.8	Редагування профілю
UC-3.2	Авторизація компанії та вхід до системи
UC-3.2.1	Перегляд публікованих вакансій
UC-3.2.2	Створити вакансію

Продовження таблиці 3.1.

UC-3.2.3	Видалити вакансію
UC-3.2.4	Пошук робітників
UC-3.2.5	Написати знайденому робітнику
UC-3.2.6	Написати робітнику, який прийняв вакансію.
UC-3.2.7	Перегляд чатів
UC-3.2.8	Вихід із системи
UC-4	Перегляд меню авторизації адміністратора
UC-5	Авторизація адміністратора та вхід до системи
UC-5.1	Перегляд повних профілів користувачів
UC-5.2	Редагування користувача
UC-5.3	Видалення користувача
UC-5.4	Підтвердити публікацію вакансії
UC-5.5	Вихід із системи

Всі можливі варіанти використання системи були зазначені у таблиці 3.1.

У таблицях 3.2 – 3.28 наведені детальні описи кожного варіанту використання системи.

Опис варіанту використання з шифром UC-1 наведено в таблиці 3.2.

Таблиця 3.2 – Варіант використання UC-1.

Назва	Перегляд меню авторизації
Опис	Користувач має змогу вибрати свій тип аккаунту.
Учасники	Компанія, робітник

Продовження таблиці 3.2.

Передумови	Відкритий додаток
Постумови	Відображено екран для вводу номеру телефону.
Основний сценарій	1. Користувач відкриває додаток 2. Користувач вибирає тип аккаунту
Розширення	

Опис варіанту використання з шифром UC-1 наведено в таблиці 3.3.

Таблиця 3.3 – Варіант використання UC-2

Назва	Введення номеру телефону користувача.
Опис	Користувач вводить свій номер телефону, щоб отримати смс повідомлення з кодом.
Учасники	Робітник, компанія.
Передумови	Вибраний тип аккаунту.
Постумови	Відображено екран для вводу смс коду.
Основний сценарій	1. Користувач вводить номер телефону 2. Користувач натискає кнопку «Далі»
Розширення	



Опис варіанту використання з шифром UC-3 наведено в таблиці 3.4.

Таблиця 3.4 – Варіант використання UC-3.

Назва	Введення смс коду.
Опис	Користувач вводить смс код, який надійшов йому на мобільний телефон та при правильності коду входить в систему.
Учасники	Робітник, компанія.
Передумови	Введений правильний номер телефону.
Постумови	Надано доступ до функціоналу користувача.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Користувач вводить смс код</li> <li>2. Користувач натискає кнопку «Варифікувати»</li> <li>3. При правильності коду входить до системи</li> </ol>
Розширення	Вивід про неправильність смс коду.

Опис варіанту використання з шифром UC-3.1 наведено в таблиці 3.5.

Таблиця 3.5 – Варіант використання UC-3.1.

Назва	Авторизація робітника та вхід до системи.
Опис	При правильності вводу коду проходить авторизація робітника та надається доступ до системи.
Учасники	Робітник.
Передумови	Правильно введений смс код.

Продовження таблиці 3.5

Постумови	Надано доступ до повного функціоналу робітника.
Основний сценарій	1. Робітник натискає клавішу «Варифікувати» 2. Робітник проходить авторизацію та входить до системи
Розширення	

Опис варіанту використання з шифром UC-3.1.1 наведено в таблиці 3.6.

Таблиця 3.6 – Варіант використання UC-3.1.1.

Назва	Парегляд доступних вакансій.
Опис	Робітник має змогу переглянути список підібраних для нього вакансій.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	Робітник має змогу прийняти або відхилити вакансію.
Основний сценарій	1. Робітник натискає на вкладку вакансій 2. Система відображає доступні вакансії
Розширення	

Опис варіанту використання з шифром UC-3.1.2 наведено в таблиці 3.7.

Таблиця 3.7 – Варіант використання УС-3.1.2.

Назва	Прийняти вакансію.
Опис	Робітник має змогу прийняти вакансію.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	У вкладці з чатами буде відображено чат з компанією прийнятої роботи.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Робітник натискає на вкладку вакансій</li> <li>2. Система відображає доступні вакансії</li> <li>3. Робітник приймає вакансію</li> <li>4. Система дає змогу спілкуватися з роботодавцем</li> </ol>
Розширення	

Опис варіанту використання з шифром УС-3.1.3 наведено в таблиці 3.8.

Таблиця 3.8 – Варіант використання УС-3.1.3.

Назва	Відхилити вакансію.
Опис	Робітник має змогу переглянути список підібраних для нього вакансій.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	

Основний сценарій	1. Робітник натискає на вкладку вакансій 2. Система відображає доступні вакансії 3. Робітник відхиляє вакансію
Розширення	

Опис варіанту використання з шифром UC-3.1.4 наведено в таблиці 3.9.

Таблиця 3.9 – Варіант використання UC-3.1.4.

Назва	Написати роботодавцю.
Опис	Робітник має спілкуватися з роботодавцем.
Учасники	Робітник.
Передумови	Авторизований робітник, прийнята вакансія від даної компанії.
Постумови	
Основний сценарій	1. Робітник натискає на вкладку чатів 2. Система відображає доступні чати 3. Робітник вибирає потрібний чат 4. Робітник пише та надсилає повідомлення до чату
Розширення	

Опис варіанту використання з шифром UC-3.1.5 наведено в таблиці 3.10.

Таблиця 3.10 – Варіант використання UC-3.1.5.

					ІТ61.170БАК.004 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Назва	Перегляд чатів відповідних до прийнятих вакансій.
Опис	Робітник має переглянути список чатів, які відповідають прийнятим вакансіям.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	Можливість написати в вибраний чат.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Робітник натискає на вкладку чатів</li> <li>2. Система відображає доступні чати</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.1.6 наведено в таблиці 3.11.

Таблиця 3.11 – Варіант використання UC-3.1.6.

Назва	Вихід із системи.
Опис	Робітник має змогу в будь який час вийти із системи.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	Для подальшої взаємодії треба знову авторизуватися.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Робітник відкриває меню</li> <li>2. Система відображає меню робітника</li> <li>3. Робітник натискає клавішу виходу із системи</li> </ol>

Продовження таблиці 3.11

Розширення	
------------	--

Опис варіанту використання з шифром UC-3.1.7 наведено в таблиці 3.12.

Таблиця 3.12 – Варіант використання UC-3.1.7.

Назва	Перегляд профілю.
Опис	Робітник має змогу переглянути свій профіль.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	Робітник має змогу редагувати профіль.
Основний сценарій	1. Робітник відкриває вкладу з профілем 2. Система відображає профіль робітника
Розширення	

Опис варіанту використання з шифром UC-3.1.8 наведено в таблиці 3.13.

Таблиця 3.13 – Варіант використання UC-3.1.8.

Назва	Редагування профіль.
Опис	Робітник має змогу редагувати профіль.
Учасники	Робітник.
Передумови	Авторизований робітник.
Постумови	

Продовження таблиці 3.13.

					IT61.170БАК.004 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Основний сценарій	<ol style="list-style-type: none"> <li>1. Робітник відкриває вкладу з профілем</li> <li>2. Система відображає профіль робітника</li> <li>3. Робітник натискає клавішу редагування профілю</li> <li>4. Робітник редагує параметри профілю та зберігає зміни</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.2 наведено в таблиці 3.14.

Таблиця 3.14 – Варіант використання UC-3.2.

Назва	Авторизація компанії та вхід до системи.
Опис	При правильності вводу коду проходить авторизація компанії та надається доступ до системи.
Учасники	Компанія.
Передумови	Правильно введений смс код.
Постумови	Надано доступ до повного функціоналу компанії.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає кнопку «Варифікувати»</li> <li>2. Компанія проходить авторизація та має доступ до повного функціоналу</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.2.1 наведено в таблиці 3.15.

Таблиця 3.15 – Варіант використання UC-3.2.1.

					IT61.170БАК.004 ПЗ	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Назва	Перегляд публікованих вакансій.
Опис	Компанія має змогу переглянути список публікованих вакансій.
Учасники	Компанія.
Передумови	Авторизована компанія.
Постумови	Компанія має змогу видалити вакансію.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає на вкладку з публікованими вакансіями</li> <li>2. Система відображає публіковані вакансії</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.2.2 наведено в таблиці 3.16.

Таблиця 3.16 – Варіант використання UC-3.2.2.

Назва	Створити вакансію.
Опис	Компанія має змогу створити вакансію.
Учасники	Компанія.
Передумови	Авторизована компанія.
Постумови	Адміністратор для публікації повинен верифікувати вакансію.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає кнопку «Створити вакансію»</li> </ol>

Продовження таблиці 3.16



Основний сценарій	2. Система відображає екран для створення вакансії 3. Компанія заповняє поля майбутньої вакансії та натискає кнопку створити 4. Система зберігає вакансію та відправляє її на верифікацію адміністратору
Розширення	

Опис варіанту використання з шифром UC-3.2.3 наведено в таблиці 3.17.

Таблиця 3.17 – Варіант використання UC-3.2.3.

Назва	Видалити вакансію.
Опис	Компанія має змогу видалити вакансію.
Учасники	Компанія.
Передумови	Авторизована компанія, створена вакансія.
Постумови	У списку вакансій для верифікації зникає видалена робота.
Основний сценарій	1. Компанія натискає на вкладку з публікованими вакансіями 2. Система відображає публіковані вакансії 3. Компанія вибирає потрібну вакансію 4. Система відображає деталі вакансії 5. Компанія натискає на кнопку «Видалити» та підтверджує свій вибір

Продовження таблиці 3.17.

Основний сценарій	6. Система видаляє вакансію
Розширення	

Опис варіанту використання з шифром UC-3.2.4 наведено в таблиці 3.18.

Таблиця 3.18 – Варіант використання UC-3.2.4.

Назва	Пошук робітників.
Опис	Компанія має змогу шукати робітників серед зареєстрованих в системі.
Учасники	Компанія.
Передумови	Авторизована компанія.
Постумови	Система відобразить список знайдених робітників та компанія має змогу продивитися профілі та написати робітнику.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає на вкладку пошуку</li> <li>2. Система відображає екран для заповнення параметрів пошуку</li> <li>3. Компанія заповняє параметри та натискає клавішу «Шукати»</li> <li>4. Система шукає робітників та виводить їх на екран</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.2.5 наведено в таблиці 3.19.

					ІТ61.170БАК.004 ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 3.19 – Варіант використання УС-3.2.5.

Назва	Написати знайденому робітнику.
Опис	Компанія має змогу написати знайденому робітнику або подзвонити через аудіо-зв'язок.
Учасники	Компанія.
Передумови	Авторизована компанія, ініційований пошук.
Постумови	У робітника в списку чатів з'явиться новий чат.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає на вкладку пошуку</li> <li>2. Система відображає екран для заповнення параметрів пошуку</li> <li>3. Компанія заповняє параметри та натискає клавішу «Шукати»</li> <li>4. Система шукає робітників та виводить їх на екран</li> <li>5. Компанія натискає клавішу «Написати» та погоджується зняти кошти з балансу</li> <li>6. Система відображає чат</li> <li>7. Компанія вводить повідомлення та натискає клавішу відправити</li> </ol>
Розширення	

Опис варіанту використання з шифром УС-3.2.6 наведено в таблиці 3.20.

Таблиця 3.20 – Варіант використання УС-3.2.6.

					ІТ61.170БАК.004 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Назва	Написати робітнику, який прийняв вакансію.
Опис	Компанія має змогу спілкуватися з робітником, який прийняв публіковану вакансію або подзвонити по аудіо-зв'язку.
Учасники	Компанія.
Передумови	Авторизована компанія, створена вакансія, вакансія прийнята робітником.
Постумови	
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає на вкладку з чатами</li> <li>2. Система відображає список доступних чатів</li> <li>3. Компанія натискає на вибраний чат</li> <li>4. Система відображає чат</li> <li>5. Компанія набирає повідомлення та натискає кнопку «Відправити»</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.2.7 наведено в таблиці 3.21.

Таблиця 3.21 – Варіант використання UC-3.2.7.

Назва	Перегляд чатів.
Опис	Компанія має змогу переглянути чати з робітниками, які прийняли публіковані вакансії.
Учасники	Компанія.

Продовження таблиці 3.21

Передумови	Авторизована компанія.
Постумови	Компанія матиме змогу написати робітнику.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія натискає на вкладку з чатами</li> <li>2. Система відображає доступні чати</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-3.2.8 наведено в таблиці 3.22.

Таблиця 3.22 – Варіант використання UC-3.2.8.

Назва	Вихід із системи.
Опис	Компанія має змогу в будь-який час вийти із системи.
Учасники	Компанія.
Передумови	Авторизована компанія.
Постумови	Для доступу до функціоналу потрібно буде авторизуватися.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Компанія відкриває меню</li> <li>2. Компанія натискає кнопку «Вийти»</li> <li>3. Система повертається до меню авторизації</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-4 наведено в таблиці 3.23.

Таблиця 3.23 – Варіант використання УС-4.

Назва	Меню авторизації адміністратора
Опис	Адміністратор на окремому веб-додатку має змогу авторизуватися та увійти у систему.
Учасники	Адміністратор.
Передумови	
Постумови	При успішній авторизації адміністратор матиме доступ до функціоналу.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Адміністратор відкриває сайт</li> <li>2. Адміністратор заповняє логін і пароль та натискає «Увійти»</li> </ol>
Розширення	

Опис варіанту використання з шифром УС-5 наведено в таблиці 3.24.

Таблиця 3.24 – Варіант використання УС-5.

Назва	Доступ до повного функціоналу адміністратора.
Опис	Після успішної авторизації адміністратор має доступ до повного функціоналу.
Учасники	Адміністратор.
Передумови	Успішна авторизація.
Постумови	
Основний сценарій	<ol style="list-style-type: none"> <li>1. Адміністратор натискає кнопку «Увійти»</li> </ol>

Продовження таблиці 3.24

Основний сценарій	2. Система авторизує адміністратора та надає доступ до функціоналу
Розширення	

Опис варіанту використання з шифром UC-5.1 наведено в таблиці 3.25.

Таблиця 3.25 – Варіант використання UC-5.1.

Назва	Перегляд повних профілів користувачів.
Опис	Адміністратор має змогу переглянути профілі.
Учасники	Адміністратор.
Передумови	Успішна авторизація.
Постумови	
Основний сценарій	1. Адміністратор натискає на вкладку з користувачами 2. Система виводить список користувачів
Розширення	

Опис варіанту використання з шифром UC-5.2 наведено в таблиці 3.26.

Таблиця 3.26 – Варіант використання UC-5.2.

Назва	Редагувати користувача.
Опис	Адміністратор має можливість редагувати профілі користувачів, а також видаляти вакансії у компаній.

Продовження таблиці 3.26

Учасники	Адміністратор.
Передумови	Успішна авторизація.
Постумови	
Основний сценарій	<ol style="list-style-type: none"> <li>1. Адміністратор натискає на вкладку з користувачами</li> <li>2. Система виводить список користувачів</li> <li>3. Адміністратор вибирає потрібного користувача, вносить зміни та зберігає</li> </ol>
Розширення	

Опис варіанту використання з шифром UC-5.3 наведено в таблиці 3.27.

Таблиця 3.27 – Варіант використання UC-5.3.

Назва	Підтвердити публікацію вакансії.
Опис	Перш ніж вакансія буде активна для робітників, адміністратор повинен верифікувати її.
Учасники	Адміністратор.
Передумови	Успішна авторизація, заявка підтвердження вакансії.
Постумови	Вакансія пропонується робітникам.
Основний сценарій	<ol style="list-style-type: none"> <li>1. Адміністратор натискає на з не підтвердженими вакансіями</li> <li>2. Система виводить список вакансій</li> </ol>



Продовження таблиці 3.27

Основний сценарій	3. Адміністратор підтверджує вакансію
Розширення	

Опис варіанту використання з шифром UC-5.4 наведено в таблиці 3.28.

Таблиця 3.28 – Варіант використання UC-5.4.

Назва	Вихід із системи адміністратора.
Опис	Адміністратор має змогу в будь-який час вийти із системи.
Учасники	Адміністратор.
Передумови	Успішна авторизація.
Постумови	Для використання функціоналу потрібно авторизуватися знову
Основний сценарій	<ol style="list-style-type: none"> <li>1. Адміністратор відкриває меню та натискає кнопку «Вийти»</li> <li>2. Система переходить до меню авторизації</li> </ol>
Розширення	

### 3.2 Розробка функціональних вимог до системи

Маса проблем із програмним забезпеченням виникає через недосконалість способів, які люди застосовують для збору, документування, узгодження модифікації вимог до ПО. Проблеми можуть виникати через неформального збору інформації, передбачуваної функціональності, помилкових або

					IT61.170БАК.004 ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

неузгоджених припущень, недостатньо визначених вимог і безсистемного зміни процесу. [4]

Функціональні вимоги відповідні до варіантів використання системи:

- відображення меню авторизації для користувачів;
- авторизація за допомогою смс коду, надісланому на телефон;
- відображення публікованих вакансій компанії;
- відображення доступу вакансій для робітника;
- створення вакансії компанією;
- видалення вакансії компанією;
- прийом вакансії робітником;
- перегляд чатів;
- спілкування за допомогою чатів та аудіо-дзвінків;
- перегляд профілю;
- редагування профілю;
- пошук робітників;
- перегляд повних профілів адміністратором;
- редагування профілів адміністратором;
- верифікувати роботу адміністратором;
- вихід із системи.

### 3.3 Розробка нефункціональних вимог до системи

Як правило, кажучи про нефункціональних вимогах, найчастіше говорять про атрибути якості (тобто вимогах, що визначають якісні характеристики розроблюваного програмного забезпечення або системи, такі як продуктивність, надійність, масштабованість), не звертаючи уваги на інші види нефункціональних вимог, а саме : обмеження, бізнес-правила, зовнішні інтерфейси, пропозиції щодо реалізації, юридичні вимоги. [5]

					IT61.170БАК.004 ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

Проаналізувавши предметну область та оглянувши існуючі рішення можна скласти список не функціональних вимог, які повинні бути присутні у системі:

- розширюваність – система повинна бути готова до внесення змін, додавання нового функціоналу і не ламатися;
- переносимість – система повинна мати змогу розгортатися на різних платформах;
- перевикористання компонентів – код повинен бути організований так чином, щоб модулі можна було використовувати в різних частинах коду без повторного описання. Це дуже економить час на розробці;
- автоматизоване тестування – система повинна мати написані автоматизовані тести щоб при розробці нового функціоналу він не призводив до помилок в існуючому коді;
- безпека – один з найголовіших факторів. Система має використовувати останні технології безпеки, бути стійкою до атак, зберігати данні у шифрованому вигляді та обмінюватися даними безпечними протоколами.

### 3.4 Висновок до розділу

У даному розділі за допомогою проаналізованої інформації з минулих розділів було створено діаграму варіантів використання, також були сформульовані функціональні та не функціональні вимоги до системи.

Варто зазначити, що такий підхід до створення системи полегшує майбутню розробку та тестування.

					ІТ61.170БАК.004 ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

## 4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

### 4.1 Переваги вибраних технологій розробки

Дійсно, вибір нового програмного забезпечення є досить повільним процесом, коли понад 50% роботи робиться ще на етапі аналізу попередніх вимог. Це пов'язано з основними вимогами сучасного світу – високим ступенем автоматизації і швидкою адаптацією процесів і систем до потреб клієнтів і ринку. Відповідно, використовувані при створенні програмного забезпечення технології повинні забезпечувати високу стабільність, гарантовану спадкоємність протягом тривалого проміжку часу, масштабованість і, що важливо, повторюваність і стійкість інфраструктури. [6]

Розробка будь-якого застосунку не обмежується тільки однією мовою програмування чи фреймворків. Клієнт-серверна архітектура передбачає як мінімум 2 головних програми (сервер і клієнтська частина), тому є можливість писати на декількох мовах. А оскільки серверна частина може складатися з декількох модулів, то навіть вони можуть бути написані на різних мовах програмування.

Вибір стеку технологій залежить від багатьох факторів пов'язаних з системою та командою розробки.

По-перше, кожен програміст має вподобання та досвід в роботі з певними мовами, фреймворками та середовищами розробки, тому фактор навичок команди буде сильно впливати на швидкість та якість розробки.

По-друге, кожна мова та фреймворк мають свої особливості роботи та рекомендовані в певних сферах розробки, тому не менш важливим фактором являються особливості та вимоги системи. Наприклад швидкість виконання, сумісність з апаратною частиною і тд.

Варто зазначити, що ще одним важливим фактором є фінансова сторона, оскільки за роботу сервера, сторонні сервіси та підтримку апаратної частини треба платити гроші.

					IT61.170БАК.004 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

Проект був створений за допомогою мови JavaScript, тому і фреймворки були вибрані відповідні. Отож, при розробці автоматизованої платформи пошуку роботи було використано такі технології:

- мова програмування JavaScript;
- серверна платформа Node.js;
- база даних MongoDB;
- фреймворк для нативних мобільних додатків React Native;
- бібліотека для роботи чату та аудіо дзвінків на веб сокетах Sock.io;
- бібліотека стилізованих компонентів Material UI;
- функціональна бібліотека для JavaScript – Ramda;
- веб-сервіс зберігання проектів GitHub.

Також було використана велика кількість інших бібліотек, як допоміжних для покращення роботи системи.

#### 4.1.1 JavaScript

JavaScript – це мультипарадигменна мова програмування, що підтримує керування подіями, функціональні та імперативні стилі програмування. Вона має інтерфейси прикладного програмування (API) для роботи з текстом, датами, регулярними виразами, стандартними структурами даних та моделлю об'єкта документа (DOM).[7]

Мова JavaScript був розроблений Бренданом Айком з корпорації Netscape Communications Corporation в 1995 році. Його перша розробка була вельми скоростиглої, і критики JavaScript здебільшого засуджували недоліки попереднього планування під час його розробки. Однак Брендан Аїк ні дилетантом: у нього був серйозний досвід в інформатиці, і він заклад в JavaScript на подив складні і передові ідеї. Так чи інакше, він випередив час, і зажадав 15 років, щоб цей чудовий мову завоював популярність у провідних розробників. [8]

					IT61.170БАК.004 ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Також на мові JavaScript розробляються мікроконтроллери, мобільні та настільні додатки, серверні частини.

Переваги мови JavaScript:

- швидкість виконання, JavaScript запускається за допомогою веб браузерів;
- простота в розумінні коду та синтаксису;
- популярність, оскільки JavaScript дуже популярний – багато посібників, бібліотек та форумів можна знайти на просторах Інтернету;
- незалежність, деякі програми на JavaScript можуть функціонувати і без серверної частини;
- багатофункціональні інтерфейси;
- універсальність;
- підтримка, специфікація для JavaScript завжди знаходиться в розробці та кожен рік виходить нова версія.

У проєкті буде використана ця мова, оскільки універсальність цієї мови дозволяє розробляти як серверні частини, так і мобільні додатки.

#### 4.1.2 Node.js

Node.js - це платформа, побудована на JavaScript для зручного створення швидких та масштабованих мережевих додатків. Node.js використовує модель керування подіями, що не блокує введення / виведення та робить її легкою та ефективною, ідеальною для використання додатків в реальному часі, які працюють на розподілених пристроях. [9]

З моменту появи Node.js в 2009 році мову JavaScript пройшов довгий шлях від ледве стерпного браузерного мови до одного з найважливіших мов в усіх областях розробки програмного забезпечення. Почасти ця зміна пов'язана з появою специфікації ECMAScript 2015 року, усунувши ряд важливих недоліків в попередніх версіях мови. Node використовує JavaScript-ядро Google V8,

					IT61.170БАК.004 ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

засноване на шостої версії стандарту ECMAScript (іноді вона називається ES6 і позначається скороченням ES2015). Також на ситуацію вплинули такі інноваційні технології, як Node, React і Electron; вони дозволяють застосовувати JavaScript буквально повсюдно: від сервера до браузера і в платформних мобільних додатках. Найбільші компанії поступово беруть JavaScript, а компанія Microsoft навіть внесла свій вклад в успіх Node. [10]

Переваги платформи Node.js:

- асинхронність та модель керування подіями - всі API бібліотеки Node.js асинхронні, тобто не блокуються. Це по суті означає, що сервер на базі Node.js ніколи не чекає повернення даних API. Після виклику сервера переходить до наступного API, а механізм сповіщень про події Node.js допомагає серверу отримати відповідь від попереднього виклику API;
- швидкий у виконанні завдяки JavaScript;
- однопоточний, але дуже масштабований - Node.js використовує однопоточну модель з циклом подій. Механізм подій допомагає серверу реагувати без блокування інших подій і робить сервер високо масштабованим на відміну від традиційних серверів, які створюють обмежені потоки для обробки запитів;
- без буферизації.

#### 4.1.3 MongoDB

MongoDB – це міжплатформна програма, орієнтована на документи. MongoDB, класифікований як програма баз даних NoSQL, використовує документи, подібні JSON, зі схемою.[11]

Вибір бази даних дуже важлива частина проектування. Вибір бази даних впливає на фінансові витрати, швидкість оброблення даних та складність масштабування. Тому варто обирати уважно. В іншому випадку вибір неправильної БД може стати перешкодою, коли проект росте і іноді неможливо

					IT61.170БАК.004 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

виправити. Навіть якщо ви працюєте над зрілим проектом, який певний час використовує конкретний БД, важливо знати його обмеження та визначити, коли слід додати інший тип БД у свій стек (дуже часто зустрічається декілька БД).

Переваги MongoDB:

- проста в установці та налаштуванні;
- не має схем для об'єктів, схему формує код програми;
- динамічні запити;
- проста у масштабуванні;
- захищена, немає можливості зробити SQL ін'єкцію.

#### 4.1.4 React Native

React Native – це сценарій JavaScript для написання мобільних додатків для iOS та Android. Він заснований на React, бібліотеці JavaScript Facebook для побудови інтерфейсів користувача, але замість націлення на браузер, він націлений на мобільні платформи. Крім того, оскільки більшість написаних вами кодів можна поділити між платформами, React Native полегшує одночасну розробку і для Android, і для iOS. [12]

Переваги React Native:

- необов'язкове виконання частин коду;
- повторне використання готових компонентів;
- популярність;
- можливість використання сторонніх плагінів;
- підтримка «гарячого» перезавантаження;
- модульна архітектура;
- написання коду одночасно для декількох платформ.

					IT61.170БАК.004 ПЗ	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		



#### 4.1.5 Socket.io

Socket.io – це бібліотека, яка забезпечує зв'язок між браузером та сервером у режимі реального часу, двонаправленим та подієвим. [13]

Socket.IO в основному використовує протокол WebSocket з опитуванням як резервний варіант, забезпечуючи той самий інтерфейс. Хоча він може бути використаний як просто обгортка для WebSocket, він надає багато інших функцій, включаючи трансляцію в декілька сокетів, зберігання даних, пов'язаних з кожним клієнтом, і асинхронний введення / виведення.

Переваги Socket.IO:

- надійність;
- підтримка автоматичного відновлення зв'язку;
- виявлення відключення;
- підтримка кімнат.

#### 4.1.6 Material UI

Material UI – це бібліотека з готовими стилізованими компонентами для прискорення розробки інтерфейсу користувача. Готові компоненти мають параметри, за допомогою яких можна налаштовувати їх під проект.

Переваги Material UI:

- швидкість розробки;
- зручне налаштування;
- приємний дизайн;

В проекті ця бібліотека буде задіяна частково враховуючи особливості сторінок та дизайну мобільного додатку.

					ІТ61.170БАК.004 ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

#### 4.1.7 Ramda

Уже є кілька чудових бібліотек з функціональним колоритом. Зазвичай вони призначені для набору інструментів загального призначення, придатних для роботи в декількох парадигмах. Ramda має більш цілеспрямовану мету. Ми хотіли, щоб бібліотека була розроблена спеціально для функціонального стилю програмування, яка спрощує створення функціональних конвеєрів, та ніколи не мутує дані користувачів. Оскільки проект буде написаний в функціональному стилі, ця бібліотека спростить написання такого коду.

Переваги Ramda та функціонального стилю:

- чисті функції легше читати;
- краще тестування через чисті функції та перевірку лише входів та виходів;
- потрібно менше коду для написання;
- більше коду для повторного використання.

#### 4.2 Висновок до розділу

У цьому розділі на основі аналізу предметної області, огляду існуючих рішень, поставлених вимог до системи та особистого досвіду було сформовано список технологій, які використані при розробці проекту.

					ІТ61.170БАК.004 ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 РОЗРОБКА СИСТЕМИ

### 5.1 Структура проекту

#### 5.1.1 Структура серверної частини проекту

На рисунку 5.1 зображено структуру серверної частини проекту.

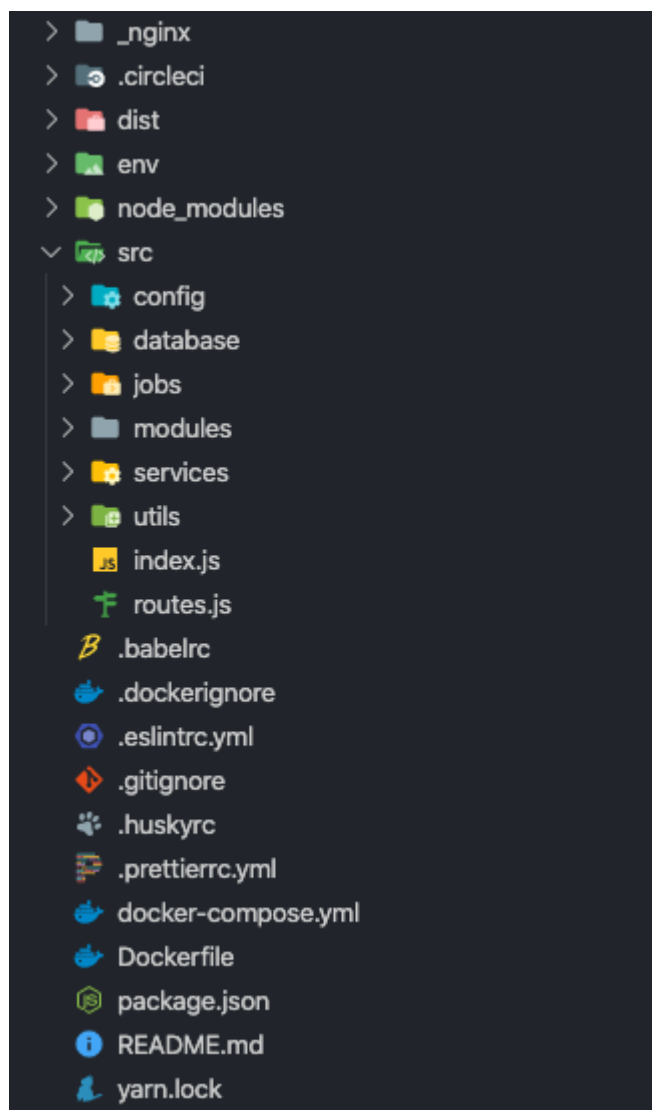


Рисунок 5.1 – Структура серверної частини проекту

Структура серверної частини проекту складається з наступних компонентів:

- `_nginx` – конфігурації платформи (веб серверу);

					IT61.170БАК.004 ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

- circlesi – конфігурації платформи для безперервної інтеграції коду та автоматизації тестування;
- dist – збірка готово додатку;
- env – конфіденційні константи додатку (ключі, токени, паролі);
- node\_modules – файли зовнішніх бібліотек;
- config – конфігурації та константи, які програмно оброблені;
- database – налаштування бази даних, міграції;
- jobs – перелік подій, які виконуються з затримкою (сповіщення, архівація);
- modules – основна папка проекту. У ній знаходяться модулі, які відповідають за кожну модель та містять контроллер, модель та роути;
- services – описані класи сторонніх сервісів, які використовуються у проекті;
- utils – готові функції, які часто використовуються;
- index – головний файл проекту, з якого починається запуск програми;
- routes – перелік усіх роутів проекту;
- babelrc – конфігурації JavaScript компілятора;
- .dockerignore – містить перелік файлів, які не включаються до зборки контейнера Docker;
- .eslintrc.yml – файл конфігурації JavaScript літера Eslint;
- .gitignore – список файлів, резвізію яких Git не проводить;
- .huskyrc – конфігурації для перевірки коду перед створенням коміту;
- .prettierrc.yml – файл конфігурації для літера Prettier;
- docker-compose.yml – містить конфігурації Docker контейнеру;
- Dockerfile – містить скрипт для запуск Docker контейнеру;
- package.json – перелік сторонніх бібліотек та команд взаємодії з програмою;
- README.md – вказівки розробника;
- yarn.lock – містить детальний опис використаних бібліотек.

					IT61.170БАК.004 ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

### 5.1.2 Структура клієнтської частини проекту

На рисунку 5.2 зображено структуру мобільного додатку.

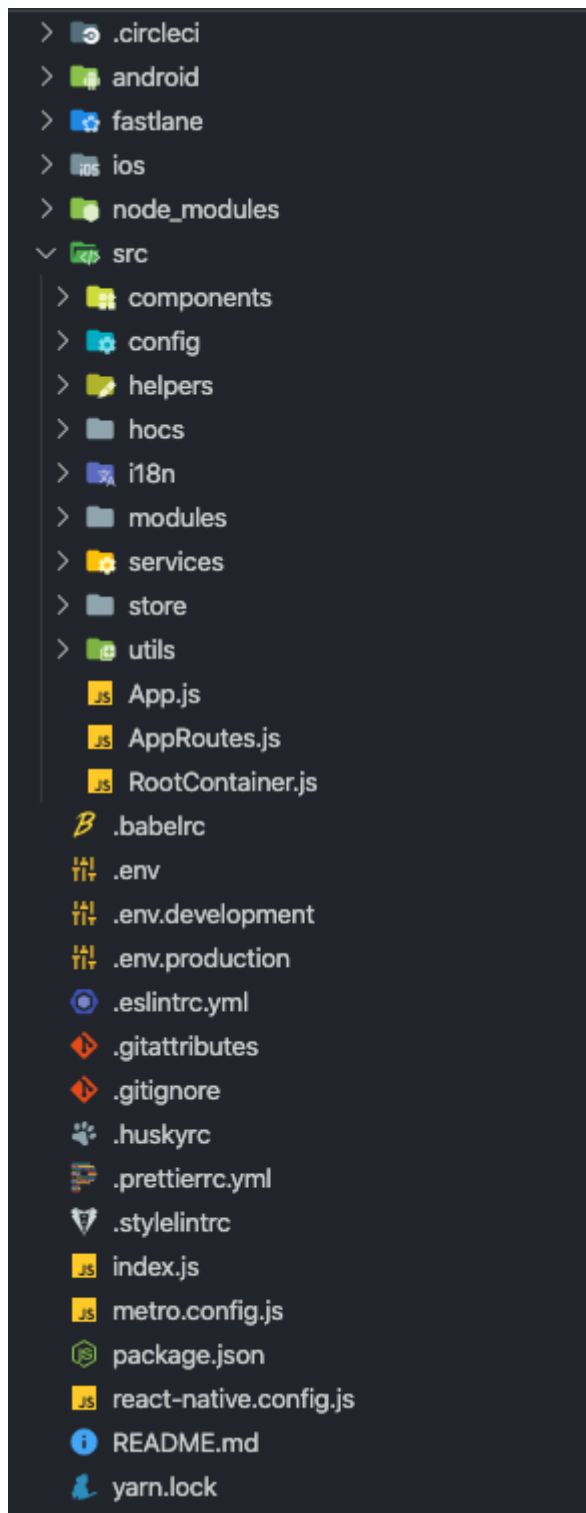


Рисунок 5.2 Структура клієнтської частини проекту

					IT61.170БАК.004 ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура серверної частини проекту складається з наступних компонентів:

- circlesi – конфігурації платформи для безперервної інтеграції коду та автоматизації тестування;
- android – містить конфігурації для збірки додатку на Android смартфонах;
- fastlane – конфігурації для оновлення версій додатку на платформах Google Play та AppStore;
- ios – конфігурації для збірки додатку на платформі IOS;
- node\_modules – файли зовнішніх бібліотек;
- components – React компоненти, які часто використовуються у різних модулях;
- config – конфігурації та константи, які програмно оброблені;
- helpers – функції, які призначені для зменшення коду обробки інформації у компонентах та кожна з них має спеціальне призначення;
- hocs – компоненти вищого порядку, які містять універсальний функціонал. В них можна обгорнути будь-який компонент для отримання функціоналу;
- i18n – модуль, який відповідає за мову тексту відображеного на інтерфейсі користувача;
- modules – містить модулі, які відповідають за роботу одного окремого функціоналу та становить основну частину написаного коду;
- services – описані класи сторонніх сервісів, які використовуються у проекті;
- store – містить конфігурації для глобального сховища клієнтської програми Redux;
- utils – містить функції широкого призначення, які часто використовуються у програмі;
- App.js – містить основу розмітки та її конфігурації;

					IT61.170БАК.004 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

- AppRoutes.js – містить роутинг всіх можливих екранів додатку;
- RootContainer.js – налаштовує роути додатку та підключає їх;
- .babelrc – конфігурації JavaScript компілятора;
- .env – константи, ключі та паролі;
- .eslintrc.yml – конфігурації лінера ESLint;
- .gitattributes – конфігурації системи контролю версій Git;
- .gitignore – список файлів, які Git не включає для ревізій;
- .huskyrc – конфігурації для перевірки правильності коду перед комітом;
- .prettierrc.yml – конфігурації лінера Prettier;
- .stylelintrc – конфігурації перевірки правильності написання CSS стилів у коді;
- index.js – файл, з якого починається запуск програми;
- metro.config.js – конфігурації для завантаження JavaScript-bundle;
- package.json – перелік сторонніх бібліотек та команд взаємодії з програмою;
- react-native.config.js – конфігурації фреймворку React-Native;
- README.md – вказівки розробника щодо експлуатації;
- yarn.lock – містить детальний опис використаних бібліотек.

## 5.2 Структура обробки запитів

Оскільки система має клієнт-серверну архітектуру, то вона має збірну системи обробки інформації, тобто обмінюється даними. У розробленому кресленнику ІТ61.170БАК.004 Д2 наведено відношення між програмними компонентами (діаграма компонентів).

Також у розробленому кресленнику ІТ61.170БАК.004 Д4 наведено діаграму послідовностей головного функціоналу проекту.

Система використовує для обробки даних три головні компоненти:

					ІТ61.170БАК.004 ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

- мобільний застосунок;
- веб сервер;
- базу даних.

Веб сервер та база даних піднімається за допомогою ізольованих Docker контейнерів для полегшення розгортання системи на різних платформах.

Веб сервер являє собою посередником між інтерфейсом користувача та базою даних. Служить для обробки основних логічних процесів, внесення змін до бази даних та для взаємодії зовнішніх сервісів.

Працює веб сервер наступним чином. Сервер очікує запит, користувач за інтерфейсу виконує якусь дію. Мобільний додаток в залежності від дії надсилає запит по HTTPS або Socket з'єднанню. Якщо було надіслано HTTPS запит, то сервер за допомогою Node.js виконує потрібні дії (вносить зміни до бази даних, спілкується з зовнішніми сервісами) та підправляє відповідь до користувача. Якщо запит був відправлений по Socket з'єднанню, наприклад надіслане повідомлення в чаті, то запит надходить на Socket сервер (в даному проекті він включений до Node.js інфраструктури). Socket сервер обробляє запит та надсилає до користувача відповідь.

Варто зазначити, що Socket сервер обробляє дані в реальному часі та підтримує безперервний зв'язок з мобільним додатком. Це зроблено для швидкого обміну даними. Такий зв'язок зазвичай використовують у чатах, відео та аудіо зв'язках та відеоіграх.

### 5.3 Розробка бази даних

Оскільки у проекті використовується MongoDB, найпопулярнішою бібліотекою для взаємодії Node.js і MongoDB являється mongoose, який був використаний при розробці. За допомогою mongoose в коді створюються таблиці, автоматично при запуску додаються до бази даних, якщо їх не має. Також він слугує як DAL (data access layer) у проекті, що дозволяє зручно

					IT61.170БАК.004 ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		



взаємодіяти з кожної моделлю та дозволяє створювати свої віртуальні методи, які закріплюються за кожною таблицею окремо.

Діаграму структури бази даних наведено у розробленому кресленику IT61.170БАК.004 ДЗ. Опис і призначення таблиць наведено у таблицях 5.1-5.10.

Опис таблиці workers наведено в таблиці 5.1.

Таблиця 5.1 – Таблиця бази даних workers.

Назва таблиці	workers
Опис	<p>Таблиця зберігає дані про робітника, такі як:</p> <ul style="list-style-type: none"><li>- <code>_id</code> – ідентифікатор робітника;</li><li>- <code>phone</code> – номер користувача;</li><li>- <code>email</code> – електронна пошта;</li><li>- <code>isVerified</code> – чи верифікований робітник;</li><li>- <code>isOnboarded</code> – чи робітник пройшов реєстрацію;</li><li>- <code>profile.name</code> – ім'я робітника;</li><li>- <code>profile.jobTitle</code> – рівень робітника;</li><li>- <code>profile.driverLicense</code> - наявність водійських прав;</li><li>- <code>profile.avatar</code> – аватарка робітника;</li><li>- <code>profile.locationAddress</code> – адрес робітника;</li><li>- <code>profile.location</code> – точка робітника на карті;</li><li>- <code>profile.certifications</code> – сертифікати наявні у робітника;</li><li>- <code>profile.trades</code> – категорія та навички, якими володіє робітник.</li></ul>

Опис таблиці job\_invites наведено в таблиці 5.2.

Таблиця 5.2 – Таблиця бази даних job\_invites.

Назва таблиці	job_invites
Опис	<p>Таблиця зберігає дані про запрошення на роботу, такі як:</p> <ul style="list-style-type: none"><li>- _id – ідентифікатор запрошення;</li><li>- job_id – ідентифікатор роботи;</li><li>- chat_id – ідентифікатор чату;</li><li>- worker_id – ідентифікатор робітника;</li><li>- company_id – ідентифікатор компанії;</li><li>- status – статус запрошення;</li><li>- isContact – чи був створений чат;</li><li>- isCandidateViewed – чи продивилась компанія робітника;</li><li>- declineReasons – причини відмови запрошення.</li></ul>

Опис таблиці companies наведено в таблиці 5.3.

Таблиця 5.3 – Таблиця бази даних companies.

Назва таблиці	companies
Опис	<p>Таблиця зберігає дані про компанію, такі як:</p> <ul style="list-style-type: none"><li>- _id – ідентифікатор компанії;</li><li>- phone – номер компанії;</li><li>- email – електронна пошта;</li><li>- isAdmin – чи ця компанія адміністратор;</li></ul>

	- isVerified – чи верифікована компанія;
--	--

Продовження таблиці 5.3.

Опис	<ul style="list-style-type: none"> <li>- isOnboarded – чи компанія пройшов реєстрацію;</li> <li>- profile.name – ім'я представника;</li> <li>- profile.company - назва компанії;</li> <li>- profile.address – адреса компанії;</li> <li>- profile.website – сайт компанії.</li> </ul>
------	---

Опис таблиці chats наведено в таблиці 5.4.

Таблиця 5.4 – Таблиця бази даних chats.

Назва таблиці	chats
Опис	<p>Таблиця зберігає дані про чат, такі як:</p> <ul style="list-style-type: none"> <li>- _id – ідентифікатор чату;</li> <li>- title – заголовок чату;</li> <li>- isArchived – чи архівований чат;</li> <li>- worker_id – ідентифікатор робітника;</li> <li>- company_id – ідентифікатор компанії;</li> <li>- job_id – ідентифікатор роботи;</li> <li>- messages – масив повідомлень чату.</li> </ul>

Опис таблиці chats наведено в таблиці 5.5.

Таблиця 5.5 – Таблиця бази даних trades.

Назва таблиці	trades
---------------	--------

Опис	Таблиця зберігає дані про категорію, такі як: - <code>_id</code> – ідентифікатор категорії;
------	--

Продовження таблиці 5.5.

Опис	- <code>name</code> – назва категорії; - <code>skills</code> – навички категорії; - <code>years</code> – кількість років категорії.
------	---

Опис таблиці `job_titles` наведено в таблиці 5.6.

Таблиця 5.6 – Таблиця бази даних `job_titles`.

Назва таблиці	<code>job_titles</code>
Опис	Таблиця зберігає дані про рівень робітника, такі як: - <code>_id</code> – ідентифікатор рівня; - <code>name</code> – назва рівня.

Опис таблиці `certifications` наведено в таблиці 5.7.

Таблиця 5.7 – Таблиця бази даних `certifications`.

Назва таблиці	<code>certifications</code>
Опис	Таблиця зберігає дані про сертифікат, такі як: - <code>_id</code> – ідентифікатор сертифікату; - <code>name</code> – назва сертифікату.

Опис таблиці `jobs` наведено в таблиці 5.8.

Таблиця 5.8 – Таблиця бази даних `jobs`.

Назва таблиці	<code>certifications</code>
---------------	-----------------------------

Опис	Таблиця зберігає дані про роботу, такі як: <ul style="list-style-type: none"> <li>- <code>_id</code> – ідентифікатор роботи;</li> </ul>
------	---

Продовження таблиці 5.8.

Опис	<ul style="list-style-type: none"> <li>- <code>locationAddress</code> – адреса місця роботи;</li> <li>- <code>location</code> – точка на карті роботи;</li> <li>- <code>title</code> – рівень робітника;</li> <li>- <code>wage</code> – ціна за роботу;</li> <li>- <code>typeFrameType</code> – повний чи не повний робочий день;</li> <li>- <code>trade_id</code> – ідентифікатор категорії;</li> <li>- <code>certifications</code> – масив сертифікатів;</li> <li>- <code>startDate</code> – дата початку роботи;</li> <li>- <code>isActive</code> – чи активна робота чи архівована;</li> <li>- <code>company_author</code> – ідентифікатор компанії, яка опублікувала роботу;</li> <li>- <code>status</code> – статус роботи.</li> </ul>
------	--

Опис таблиці `messages` наведено в таблиці 5.9.

Таблиця 5.9 – Таблиця бази даних `messages`.

Назва таблиці	<code>messages</code>
Опис	Таблиця зберігає дані про повідомлення, такі як: <ul style="list-style-type: none"> <li>- <code>_id</code> – ідентифікатор повідомлення;</li> <li>- <code>chat_id</code> – ідентифікатор чату;</li> <li>- <code>content</code> – наповнення повідомлення;</li> <li>- <code>type</code> – тип повідомлення;</li> </ul>

	<ul style="list-style-type: none"> <li>- author_id – ідентифікатор автора;</li> <li>- author_type – тип автора;</li> <li>- isRead – чи прочитане повідомлення.</li> </ul>
--	---

Опис таблиці migrations наведено в таблиці 5.10.

Таблиця 5.10 – Таблиця бази даних migrations.

Назва таблиці	messages
Опис	<p>Таблиця зберігає дані про виконані міграції над базою даних та має такі поля:</p> <ul style="list-style-type: none"> <li>- _id – ідентифікатор сесії міграції;</li> <li>- migrations – міграції, які були виконані протягом сесії.</li> </ul>

#### 5.4 Висновок до розділу

У даному розділі було описано структури проекту мобільного додатку та серверної частини. Були описані компоненти та надано вичерпну інформацію про їх взаємодію. Також було детально описано таблиці бази даних.

## 6 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 6.1 Фази життєвого циклу програмного забезпечення

Тестування програмного забезпечення – це трудомісткий та затратний в плані часу процес, який пропускають заради пришвидшення виходу продукту на ринок, що тягне за собою нестабільність системи. [14]

Люди хочуть створювати складні речі, які роблять наше життя простіше. Програмне забезпечення є втіленням якої складності. Однак незалежно від того, наскільки він складний, він повинен бути гнучким, простим в обслуговуванні і поліпшення. Як цього добитися? Планування кожного кроку процесу розробки програмного забезпечення є необхідною умовою для успішного продукту.

Отож для ефективної розробки був створений життєвий цикл програмного забезпечення (SDLC – Software development life cycle). Він включає в себе такі фази:

- створення ідеї проекту;
- постановка вимог;
- проектування;
- розробка;
- тестування;
- розгортання;
- обслуговування.

На етапі постановки вимог перевіряється ринок вимог, існуючі рішення та відгуки клієнтів. Після цього етапу кожен повинен мати чітке уявлення про масштаби проекту, бюджет, ресурси та термін, а також можливі ризики та вимоги щодо забезпечення якості.

Під час проектування вибираються необхідні технології, проектується база даних, та архітектуру проекту. Тобто вимоги до проекту переводять у спроектований програмний продукт.

					ІТ61.170БАК.004 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Це тривала фаза, але менш складна, ніж попередня. Використовуючи проектний документ, програмісти кодують модулі. Завдання кодування поділяються між членами команди відповідно до їхньої області спеціалізації. Передні розробники несуть відповідальність за створення інтерфейсу і зв'язок із сервером. Адміністратори бази даних додають необхідні дані до бази даних. У своїй роботі розробники використовують вказівки щодо кодування та різні інструменти для написання та реалізації коду. Кінцевим результатом цього етапу є діючий програмний продукт та документ з вихідним кодом.

Наступним етапом є тестування. Функціональне тестування: відповідність програмного забезпечення вимогам, описаним раніше.

Тестування продуктивності: Тестування спрямоване на те, щоб з'ясувати, як працює програмне забезпечення під навантаженням (його швидкість, чуйність та стабільність).

Тестування модулів: Тестування кожного модуля окремо. Якщо у будь-якого з них є недолік, розробнику потрібно повернутися назад і виправити його.

Тестування безпеки: Як впливає з назви, цей тип тестування має на меті перевірити безпеку системи.

Тестування використання: Цей тип включає тестування орієнтованих на користувача компонентів, щоб з'ясувати, чи програмне забезпечення інтуїтивно зрозуміле, просте у використанні та зрозуміле.

Якщо з'явиться якась помилка, вона передається назад команді розробників. Після виправлення помилки, команді QA необхідно повторно протестувати її. Забезпечення якості - це постійний процес, який триває до повного позбавлення програмного забезпечення від помилок та не відповідає вимогам.

Останнім етапом являється розгортання. Коли програмне забезпечення завершено і не має помилок, воно поставляється на ринок для бета-тестування. Команда підтримки збирає відгуки перших користувачів, і якщо з'являються якісь помилки, команда розробників виправляє їх. Після цього розгортається

					IT61.170БАК.004 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		



остаточний варіант. Оновлення програмного забезпечення реалізуються на етапі технічного обслуговування, щоб переконатися, що воно є толерантним до порушень безпеки.

## 6.2 Тестування програмного забезпечення

Тестування програмного забезпечення визначається як діяльність з перевірки відповідності фактичних результатів очікуваним результатам та забезпечення того, що програмна система є бездефектною. Він передбачає виконання програмного компонента або системного компонента для оцінки однієї або декількох властивостей, що цікавлять. Тестування програмного забезпечення також допомагає виявити помилки, прогалини чи відсутні вимоги всупереч фактичним вимогам. Це можна зробити вручну або за допомогою автоматизованих інструментів. Основним критерієм вибору є вимоги для проекту, який обумовлюється тривалістю та складністю проекту. Залежність між тривалістю проекту та коефіцієнту рівня прибутковості або збитковості проведення тестів відображено на рисунку 6.1.

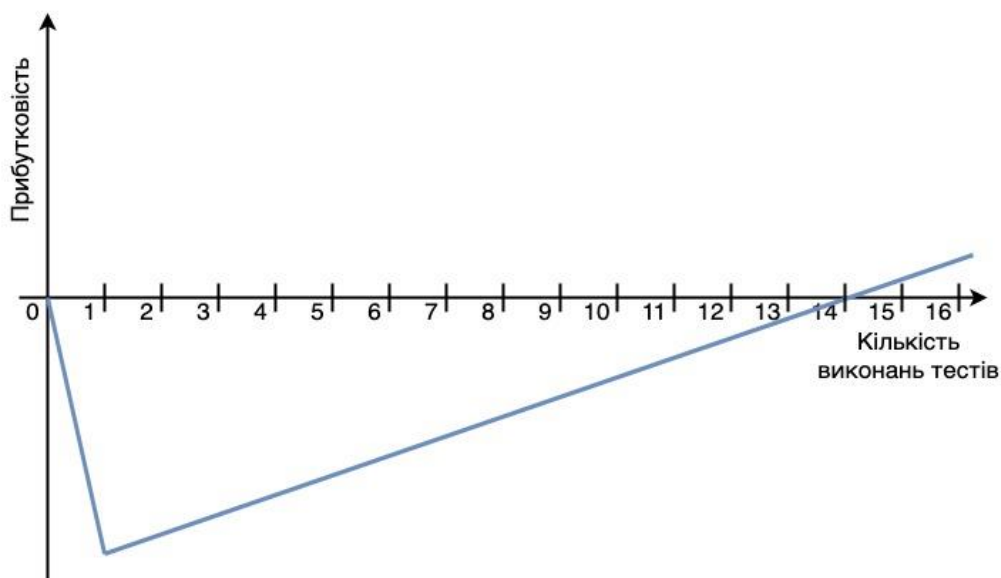


Рисунок 6.1 – Графік залежності між прибутковістю та кількістю виконання тестів

### 6.2.1 Ручне тестування

При розробці даного програмного забезпечення ручне тестування було використано переважно для виявлення помилок у коді. Для тестування в такому режимі були задіяні найпопулярніші методики. До них належать:

- метод логування, коли вставляється оператор виводу в консоль проміжних значень змінних;
- метод покрокового виконання програми, який дозволяє поетапно переглядати поточні стани всіх змінних в вибраному проміжку;
- метод точок зупинки, яка дозволяє зупиняти виконання програми у потрібному місці для перегляду стану змінних.

### 6.2.2 Автоматизоване тестування

Автоматичний тестування передбачає використання інструменту автоматизації для виконання набору тестових випадків. Навпаки, ручне тестування проводиться людиною, що сидить перед комп'ютером, ретельно виконуючи етапи тестування.

Програмне забезпечення для автоматизації також може вводити дані тестування в тестувальну систему, порівнювати очікувані та фактичні результати та створювати детальні звіти про тести. Тест-автоматизація вимагає значних вкладень грошей та ресурсів.

Для структуризації та спрощення автоматизованих тестів складається план тестування, в якому описують тестові випадки для кожного окремого тесту.

На основі сценаріїв використання був складений план тестування та проведені тести. Тестові випадки та їх результати наведені у таблицях 6.1-6.16.

					ІТ61.170БАК.004 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 6.1 – Тестований випадок «Перевірка відображення головного екрану компанії».

Дія	Авторизація компанії.
Очікуваний результат	<ul style="list-style-type: none"> <li>- успішно авторизована компанія;</li> <li>- відкрита головна сторінка компанії;</li> <li>- відображені лічильники нових прийнятих запрошень, нових повідомлень;</li> <li>- відображені кнопки для переходу на сторінку пошуку та на сторінку створення роботи.</li> </ul>
Результат	Пройшов.

Таблиця 6.2 – Тестовий випадок «Перевірка відображення головної сторінки робітника».

Дія	Авторизація робітника.
Очікуваний результат	<ul style="list-style-type: none"> <li>- успішно авторизований робітник;</li> <li>- відкрита головна сторінка робітника;</li> <li>- відображено рівень заповнення профілю, лічильники нових запрошень, повідомлень, переглядів профілю.</li> </ul>
Результат	Пройшов.

Таблиця 6.3 – Тестовий випадок «Перевірка функціонування пошуку».

Дія	Пошук робітників.
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення сторінки пошуку з формою для заповнення;</li> <li>- заповнена форма з перевіркою на правильність;</li> <li>- ініційований пошук;</li> </ul>

Продовження таблиці 6.3.

Очікуваний результат	- відображена сторінка з результатами пошуку та можливістю конвертувати пошук у публіковану вакансію.
Результат	Пройшов.

Таблиця 6.4 – Тестовий випадок «Перевірка авторизації користувача з вірними даними».

Дія	Авторизація користувача з вірними даними.
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення початкової сторінки з вибором типу аккаунту;</li> <li>- заповнена форма для номеру телефону;</li> <li>- смс повідомлення надійшло на вказаний номер;</li> <li>- вірно введений смс код;</li> <li>- відображена початкова сторінка користувача.</li> </ul>
Результат	Пройшов.

Таблиця 6.6 – Тестовий випадок «Перевірка авторизації користувача з невірними даними».

Дія	Авторизація користувача з невірними даними.
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення початкової сторінки з вибором типу аккаунту;</li> <li>- заповнена форма для номеру телефону;</li> <li>- смс повідомлення надійшло на вказаний номер;</li> <li>- відображення початкової сторінки з вибором типу аккаунту;</li> <li>- заповнена форма для номеру телефону;</li> </ul>

Продовження таблиці 6.6.

Очікуваний результат	<ul style="list-style-type: none"> <li>- смс повідомлення надійшло на вказаний номер;</li> <li>- відображення початкової сторінки з вибором типу аккаунту;</li> <li>- заповнена форма для номеру телефону;</li> <li>- смс повідомлення надійшло на вказаний номер.</li> </ul>
Результат	Пройшов.

Таблиця 6.7 – Тестовий випадок «Перевірка на створення вакансії».

Дія	Створення роботи
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення сторінки з формою для заповнення параметрів;</li> <li>- заповнена форма;</li> <li>- відображено галочки біля вірно заповнених параметрів;</li> <li>- відображення модального вікна про успішно створену вакансію;</li> <li>- в списку вакансій відображена нова.</li> </ul>
Результат	Пройшов.

Таблиця 6.8 – Тестовий випадок «Перевірка прийняття вакансії робітником»

Дія	Прийняття вакансії робітником.
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення сторінки з доступними вакансіями;</li> <li>- відображена нова вакансія;</li> <li>- натискається кнопка прийняти вакансію;</li> <li>- у компанії відображається новий робітник, який прийняв вакансію.</li> </ul>

Результат	Пройшов.
-----------	----------

Таблиця 6.9 – Тестовий випадок «Перевірка відмови вакансії робітником».

Дія	Прийняття вакансії робітником.
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення сторінки з доступними вакансіями;</li> <li>- відображена нова вакансія;</li> <li>- натискається кнопка відхили вакансію;</li> <li>- відображається запитання про причини відхилення вакансії та варіанти вибору;</li> <li>- вакансія зникає зі списку.</li> </ul>
Результат	Пройшов.

Таблиця 6.10 – Тестовий випадок «Перевірка написання повідомлення до чату».

Дія	Написання повідомлення до чату
Очікуваний результат	<ul style="list-style-type: none"> <li>- відображення сторінки з доступними вакансіями;</li> <li>- відображена нова вакансія;</li> <li>- натискається кнопка відхили вакансію;</li> <li>- відображається запитання про причини відхилення вакансії та варіанти вибору;</li> <li>- вакансія зникає зі списку.</li> </ul>
Результат	Пройшов.

Таблиця 6.11 – Тестовий випадок «Перевірка відображення списку чатів»

Дія	Відображення списку чатів
-----	---------------------------

Очікуваний результат	<ul style="list-style-type: none"> <li>- користувач натискає на вкладку з чатами;</li> <li>- відображається екран з чатами;</li> </ul>
----------------------	--

Продовження таблиці 6.11.

Очікуваний результат	<ul style="list-style-type: none"> <li>- відображається список чатів;</li> <li>- є можливість відкрити вибраний чат.</li> </ul>
Результат	Пройшов.

Таблиця 6.12 – Тестовий випадок «Перевірка відображення профілю робітника».

Дія	Відображення профілю робітника.
Очікуваний результат	<ul style="list-style-type: none"> <li>- робітник відкриває вкладку з профілем;</li> <li>- відображається профілю робітника;</li> <li>- є можливість продивитися за допомогою допоміжних вкладок всю інформацію з профілю.</li> </ul>
Результат	Пройшов.

Таблиця 6.13 – Тестовий випадок «Перевірка оновлення профілю робітника»

Дія	Оновлення профілю робітника.
Очікуваний результат	<ul style="list-style-type: none"> <li>- робітник відкриває вкладку з профілем;</li> <li>- відображається профілю робітника;</li> <li>- натискається кнопка для переходу на сторінку оновлення профілю;</li> <li>- відображається екран з різноманітними формами для оновлення профілю;</li> <li>- оновлюються вибрані поля;</li> <li>- натискається кнопка зберігання;</li> </ul>

	- відображається екран профілю робітника з оновленими даними.
Результат	Пройшов.

Таблиця 6.14 – Тестовий випадок «Перевірка перегляду профілю робітника в режимі пошуку»

Дія	Перегляд профілю робітника в режимі пошуку.
Очікуваний результат	<ul style="list-style-type: none"> <li>- натискається вкладка для переходу на форму пошуку;</li> <li>- відображення сторінки пошуку з формою для заповнення;</li> <li>- заповнена форма з перевіркою на правильність;</li> <li>- ініційований пошук;</li> <li>- відображена сторінка з результатами пошуку та можливістю конвертувати пошук у публіковану вакансію.</li> <li>- натискається кнопка для перегляду вибраного робітника;</li> </ul>

Продовження таблиці 6.14.

Очікуваний результат	- відображається профіль робітника.
Результат	Пройшов.

Таблиця 6.15 – Тестовий випадок «Перевірка конвертації форми пошуку у вакансію»

Дія	Конвертація пошуку у вакансію.
Очікуваний результат	- натискається вкладка для переходу на форму пошуку;



	<ul style="list-style-type: none"> <li>- відображення сторінки пошуку з формою для заповнення;</li> <li>- заповнена форма з перевіркою на правильність;</li> <li>- ініційований пошук;</li> </ul>
--	---

Продовження таблиці 6.15.

Очікуваний результат	<ul style="list-style-type: none"> <li>- відображена сторінка з результатами пошуку та можливістю конвертувати пошук у публіковану вакансію;</li> <li>- натискається кнопка конвертації роботи;</li> <li>- відображається форма для створення вакансії з заповненими полями;</li> <li>- додаються або змінюються поля вакансії;</li> <li>- вакансія публікується.</li> </ul>
Результат	Пройшов.

Таблиця 6.16 – Тестовий випадок «Перевірка контакту з робітником»

Дія	Контакт робітника.
Очікуваний результат	<ul style="list-style-type: none"> <li>- натискається вкладка для переходу на форму пошуку;</li> <li>- відображення сторінки пошуку з формою для заповнення;</li> <li>- заповнена форма з перевіркою на правильність;</li> <li>- ініційований пошук;</li> <li>- відображена сторінка з результатами пошуку та можливістю конвертувати пошук у публіковану вакансію;</li> <li>- компанія натискає кнопку чату на вибраному робітнику;</li> </ul>

	<ul style="list-style-type: none"> <li>- з'являється модальне вікно про трату монет на контакт з робітником;</li> <li>- відображається чат.</li> </ul>
Результат	Пройшов.

#### 6.4 Неперервна інтеграція

Випуск нових версій ПЗ пов'язаний з чималими труднощами - практично кожному розробнику доводилося проводити вихідні в офісі, займаючись розгортанням нової версії ПЗ в робочому оточенні. Такі події в кінцевому підсумку закінчуються встановленням нової версії, тому що шлях назад, до попередньої версії, у якійсь точці виявляється ще більш тернистим і небезпечним, ніж шлях вперед. Однак за установкою часто слід тривалий етап стабілізації оновленої версії. [15]

Безперервна інтеграція - це практика розробки програмного забезпечення, де члени команди часто інтегрують свою роботу, зазвичай кожна людина інтегрується щонайменше щодня, що призводить до багаторазової інтеграції на день. Кожна інтеграція перевіряється автоматизованою збіркою (включаючи тест), щоб виявити помилки інтеграції якнайшвидше. Багато команд вважають, що такий підхід призводить до значно зменшених проблем інтеграції та дозволяє команді швидше розвивати згуртоване програмне забезпечення. Для таких дій використовують готові програмні рішення для автоматизації тестів та розгортання нової версії на сервері, це економить час порівняно з виконанням вручну.

В цьому проекті для неперервної інтеграції використовується CircleCI. Точніше, як система контролю версій був задіяний GitHub, після внесення до гілки master оновлень CircleCI автоматично запускав автоматизовані тести та при успішному проходженні розгортав build на сервері.

## 6.5 Висновок до розділу

В цьому розділі було описано етапи розробки програмного забезпечення, був створений план тестування, перевірки, були створені автоматизовані тести та включені до системи неперервної розробки за допомогою CircleCI та GitHub.

					ІТ61.170БАК.004 ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

## 7 ВПРОВАДЖЕННЯ ТА ВИКОРИСТАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

### 7.1 Апаратні та програмні вимоги для експлуатації програми

Розроблена система створена за допомогою мови JavaScript. Вже на даний час JavaScript можна запустити на будь-якій із популярних операційних систем (MacOS, Linux, Windows). Node.js не споживає багато ресурсів оперативної пам'яті, основну частину об'єму жорсткого диску буде займати база даних. В таблицях 7.1-7.2 представлені мінімальні вимоги до апаратної частини сервера.

Таблиця 7.1 – Мінімальні апаратні вимоги до серверної частини.

Процесор	2-ядерний процесор з тактовою частотою 2.0 ГГц або краще.
Місце на диску	4-5 гігабайтів.
Оперативна пам'ять	3-5 гігабайти.
Операційна система	Windows 7 або пізніша, macOS, and Linux.

Таблиця 7.2 – Рекомендовані апаратні вимоги до серверної частини.

Процесор	2-ядерний процесор з тактовою частотою 2.6 ГГц або краще.
Місце на диску	10 гігабайтів.
Оперативна пам'ять	8 гігабайти.
Операційна система	Windows 7 або пізніша, macOS, and Linux.

Система створена на базі Node.js 12, тому для використання системи без неочікуваних помилок треба використовувати цю версію або новішу. Також

серверна частина реалізована за допомогою Docker 19.03.05, тому це також рекомендована опція. Також система включає наявність Інтернет зв'язку.

Клієнтською частиною виступає смартфон. Тому в таблиці 7.3 наведено мінімальні апаратні вимоги до клієнтського смартфона.

Таблиця 7.3 – Мінімальні апаратні вимоги до клієнтської частини.

Процесор	2-ядерний процесор з тактовою частотою 1.2 ГГц або краще.
Місце на диску	512 мегабайт.
Оперативна пам'ять	8 гігабайти.
Операційна система	Android 6.0, IOS 9

## 7.2 Інструкція з встановлення

Для встановлення серверної частини нам потрібно встановити Docker, який включає в себе docker-compose. Далі ми з GitHub клонуємо проект на апаратуру сервера та піднімаємо за допомогою Docker 4 контейнери.

- docker-compose run start-dependencies;
- docker-compose run --service-ports api;
- docker-compose run notification-service.

За допомогою цих команд розгортаються контейнери MongoDB, Redis, Notification Service та сам сервер.

Для встановлення мобільного додатку ми можемо викласти його за Google Play або AppStore. Встановлення на таких платформах дуже просте, треба знайти в магазині цей додаток та натиснути кнопку встановлення.

### 7.3 Висновки до розділу

У цьому розділі були сформовані мінімальні та рекомендовані вимоги для апаратної та серверної складових. Була наведена інструкція для встановлення як сервера так і застосунку.

					ІТ61.170БАК.004 ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У дипломному проекті було створено автоматизовану платформу для пошуку роботи на мові JavaScript а за допомогою бібліотек Node.js та ReactNative.

Спочатку було проведено аналіз предметної області. Були визначені призначення, цілі та особливості майбутньої системи. Також були розглянуті існуючі рішення та детально описані, вони були проаналізовані, виділені основні переваги та недоліки. На основі отриманих даних було визначено функціональні та нефункціональні вимоги до системи. Далі, на основі вимог, були вибрані технології для розробки та описано переваги використання. За допомогою всіх попередніх даних було спроектовано та розроблено базу даних, сервер та мобільний застосунок. У розділі фаз життєвого циклу розглянуто основні етапи розробки та важливість тестування. Складено план тестування, написані та успішно складені автоматизовані тести, які були прикріплені до системи неперервної розробки CircleCI.

Система готова застосування та відповідає всім поставленим вимогам та відповідає технічному завданню проекту. Також система може бути розширена за допомогою додавання або масштабування модулів модулів.

Порівняно з існуючими рішеннями розроблене програмне забезпечення має низку переваг, до яких належить: використання нових технологій, високий рівень безпеки даних, гнучке налаштування, автоматизовано більше процесів у використанні.

Розроблена автоматизована платформа пошуку роботи надає змогу розширити сферу пошуку роботи та надає клієнтам оновлену альтернативу порівняно з існуючими рішеннями.

					IT61.170БАК.004 ПЗ	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Work.ua, 2020. URL: <https://uk.wikipedia.org/wiki/Work.ua> (дата звернення: 15.05.2020).
2. Rabota.ua, 2020. URL: <https://uk.wikipedia.org/wiki/Rabota.ua> (дата звернення: 15.05.2020).
3. Jooble, 2020. URL: <https://uk.wikipedia.org/wiki/Jooble> (дата звернення: 15.05.2020).
4. Карл И. Виггерс. Разработка требований к программному обеспечению. Москва 2014. 575с.
5. Нефункциональные требования к программному обеспечению. Часть 1. 2020. URL: <https://habr.com/ru/post/231961/> (дата звернення: 16.05.2020).
6. Выбор стека технологий и баз данных для разработки корпоративного ПО, 2019. URL: <https://almexoft.com/blog/vybor-stecka-tehnologij-i-baz-dannyh-dlya-razrabotki-korporativnogo-po/> (дата звернення: 16.05.2020).
7. JavaScript. What is JavaScript, 2020. URL: <https://en.wikipedia.org/wiki/JavaScript> (дата звернення: 16.05.2020).
8. Этан Браун. Изучаем JavaScript. Руководство по созданию современных веб-сайтов. Київ, 2017. 363с.
9. Node.js What is Node.js?, 2020. URL: [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm) (дата звернення: 16.05.2020).
10. Алекс Янг, Брэдли Мек, Майк Кантелон. Node.js в действии. Санкт-Петербург, 2018. 432с.
11. MongoDB, 2018. URL: <https://en.wikipedia.org/wiki/MongoDB> (дата звернення: 16.05.2020).
12. React Native, 2020. URL: <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html> (дата звернення: 16.05.2020).

					IT61.170БАК.004 ПЗ	Арк.
						75
Змн.	Арк.	№ докум.	Підпис	Дата		



13. Socket.io. What Socket.io is. URL: <https://socket.io/docs/> (дата звернення: 16.05.2020).

14. Software Development Life Cycle: Phase and Model Guide. URL: <https://ncube.com/blog/software-development-life-cycle-guide/>. (дата звернення: 18.05.2020).

15. Емберхард Вольф. Continuous delivery. Практика непрерывных апдейтов. Санкт-Петербург, 2017. 213с.

					IT61.170БАК.004 ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		